PATENT
Attny Dkt No: 11884-400301

UNITED STATES LETTERS PATENT APPLICATION
FOR

SYSTEM AND METHOD FOR CUSTOMIZING FORM ELEMENTS IN A FORM BUILDING APPLICATION

INVENTOR:

Thomas Goering

Prepared by:

KENYON & KENYON
One Broadway
New York, NY 10004
(212) 425-7200

SYSTEM AND METHOD FOR CUSTOMIZING FORM ELEMENTS IN A FORM BUILDING APPLICATION

Copyright Notice

Background Of The Invention

[0002] Electronic forms serve an integral role in organizing information flow for today's business applications. Such forms are widely used to manage and present business data for such enterprise business applications as Customer Relationship Management (CRM), Sales and Distribution (SD), Financial Accounting (FI) and Human Resources (HR). To reduce the amount of programming skills necessary for creating and maintaining these forms, development tools have been created to enable users to design the look-and-feel of business forms in a graphical environment without coding. One such tool is the Smart Forms Form Builder application provided by SAP AG, Walldorf, Germany.

[0003] The graphical user interface (GUI) of the currently available Smart Forms Form Builder tool is depicted in FIG. 1. The GUI comprises three sections: navigation tree 100, maintenance screen 110 and form painter 120. Navigation tree 100 provides a tree structure of nodes that represent the output elements of the form such as pages, windows, graphics, tables, text areas, etc. Based on the selected node, maintenance screen 110 provides the area for inserting texts, establishing paragraph and character formats, setting up different attributes (fonts, borders, shading, etc.), or drawing tables and templates. Form painter 120 enables users to design the layout of the form.

[0004] The root nodes in navigation tree 100 are "Global Settings" and "Pages and windows". "Global Settings" has three directly inferior nodes: "Form attributes", "Form interface" and "Global definitions". Upon selection of the "Form attributes" node, maintenance screen 110 enables the user to set attributes for the entire form, such as language attributes for the translation process, page format, style and default output settings. Upon selection of the "Form interface" node, maintenance screen 110 enables the user to define the parameter interface through which the form retrieves relevant application data from an application program. And upon selection of the "Global definitions" node, maintenance screen 110 enables the user to define variables and/or constants for use throughout the form.

[0005] "Pages and windows" has two directly inferior page nodes: "FIRST" and "NEXT". Form painter 120 displays the directly inferior nodes of the "FIRST" page node, which include one graphic node ("MYSAPCOM") and four window nodes ("MAIN", "ADDRESS", "INFO" and "FOOTER"). "MAIN" includes two text nodes ("INTRODUCTION" AND "GREETINGS") and a table node ("TABLE").

[0006] Currently, users of form building applications have unrestricted access to create or edit any node in a form. However, companies may desire to restrict a user's access to particular nodes within a form for which the user is responsible. For example, one division of a company may be solely responsible for the design of the corporate logo for the form, while a second division may be responsible for defining the form interface, and a third division is assigned to process the form content.

[0007] Accordingly, there is a need in the art for a system and method that controls access to particular form elements within a form building application.

Summary Of The Invention

[0008] Embodiments of the present invention provide for customizing form elements in a form building application. According to one embodiment, a form builder receives an identification of a user of the form building application, the form building application providing a graphical user interface for the manipulation of a set of form elements, and enables access to a subset of the form elements according to authorization rules, and an access manager determines the authorization rules associated with the user.

Brief Description Of The Drawings

[0009] FIG. 1 is a screen shot of a form building application as known in the art.

[0010] FIG. 2 is a flow chart that depicts a process for reusing form elements in a form-based application runtime environment in accordance with an embodiment of the present invention.

[0011] FIG. 3 is a block diagram that depicts a client computing device in accordance with an embodiment of the present invention.

[0012] FIG. 4 is a block diagram that depicts a network architecture for a form-based application runtime environment in accordance with an embodiment of the present invention.

[0013] FIG. 5 is a block diagram of a form-based development environment in accordance with an embodiment of the present invention.

[0014] FIG. 6 is a UML class diagram for reusable form elements in accordance with an embodiment of the present invention.

Detailed Description

OVERVIEW

[0015] FIG. 2 depicts a process for implementing customizable form elements in accordance with an embodiment of the present invention. In a form-based development environment, when a form developer logs in to a form building application, the form building application first identifies the user based on the login information (step 200). With this identification information, the form building application determines the particular authorization rules associated with user (step 210), and enables access only to the form elements specified by the authorization rules (step 220).

[0016] Embodiments described below illustrate a form-based development environment within which the present invention may be implemented.

ARCHITECTURE

[0017] FIGS. 3 and 4 illustrate the components of a basic development environment in accordance with an embodiment of the present invention. FIG. 3 depicts client computing device 300, which may be a workstation, personal computer, handheld personal digital assistant ("PDA"), or any other type of microprocessor-based device. Client computing device 300 may include a processor 310, input device 320, output device 330, storage device 340, client software 350, and communication device 360.

[0018] Input device 320 may include a keyboard, mouse, pen-operated touch screen, voice-recognition device, or any other device that accepts input. Output device 330 may include a monitor, printer, disk drive, speakers, or any other device that provides output.

[0019] Storage device 340 may include volatile and nonvolatile data storage, including one or more electrical, magnetic or optical memories such as a RAM, cache, hard drive, CD-ROM drive, tape drive or removable storage disk. Communication device 360 may include a modem, network interface card, or any other device capable of transmitting and receiving signals over a network. The components of client computing device 300 may be connected via an electrical bus or wirelessly.

[0020] Client software 350 may be stored in storage device 340 and executed by processor 310, and may include, for example, the client side of a client/server application such as a form building application like Smart Forms that embodies the functionality of the present invention.

[0021] FIG. 4 illustrates a network architecture for a development environment in accordance with an embodiment of the present invention. According to one particular embodiment, when user 400a invokes a form building application, client software 350 of client computing device 300a communicates with server software 430 (e.g., the server side of the form building application) of server 420 via network link 415a, network 410, and network link 415d.

[0022] Network link 415 may include telephone lines, DSL, cable networks, T1 or T3 lines, wireless network connections, or any other arrangement that implements the transmission and reception of network signals. Network 410 may include any type of interconnected communication system, and may implement any communications protocol, which may secured by any security protocol.

[0023] Server 420 includes a processor and memory for executing program instructions, as well as a network interface, and may include a collection of servers. In one particular embodiment, server 420 may include a combination of enterprise servers such as an application server and a database server. Database 440 may represent a relational or object database, and may be accessed via a database server.

[0024] Client computing device 300 and server 420 may implement any operating system, such as Windows or UNIX. Client software 350 and server software 430 may be written in any programming language, such as ABAP, C, C++, Java or Visual Basic. Server software 440 may be built on an enterprise application platform, such as SAP Web Application Server 6.2.

FORM ELEMENT CUSTOMIZATION

[0025] Within a form-based development environment as illustrated in FIG. 5, an embodiment of the present invention may be implemented to enable access manager 520 to associate form users with authorization rules so that only the appropriate form elements are provided to the form user by form builder 510 for manipulation.

[0026] Form modeler 500 may include several components, such as form builder 510 and access manager 520, that may implement particular functionality associated with the customization of form elements. Form modeler 500 may be connected to some form of storage, such as database 440 or file system storage (local and/or remote).

[0027] According to this embodiment, form builder 510 receives identification information from a form user in the form of the user's login information to a form building application. Once the login information is received, access manager 520 references a lookup table (e.g., form definition table or file) to determine which authorization rules are associated with the user. The authorization rules may be a permission list of settings that identify a subset of form elements which are viewable and/or changeable by the user. Once the authorization rules are determined, form builder 510 may provide a graphical user interface to the user for the manipulation of only the authorized subset of form elements. This GUI may implement this functionality, for example, by only providing the authorized subset of form elements to the user, or providing all of the form elements to the user and disabling, or freezing, the unauthorized form elements. Changes to authorization rules may be made by a system administrator.

[0028] In another embodiment, form builder 510 may provide a GUI to the user that displays all form elements of a particular form. Whenever the user issues a command to change a particular form element (e.g., via selecting with input device 320 an element to be changed), access manager 520 accesses the permission list to determine whether the user is allowed to change the particular form element, and if not, rejects the user's command to change the form element (e.g., by disabling, or freezing, the unauthorized form element).

CUSTOMIZABLE FORM ELEMENT REPRESENTATION

[0029] FIG. 6 illustrates a UML class diagram representing one possible organization of customizable form elements in accordance with an embodiment of the present invention. According to this embodiment, form rules 630 includes specific authorizations for the customizable form elements. Form type 620 is a collection of form rules 630, while form class 610 is a collection of form type 620. Form 600 may include zero or one form class 610 or form type 620.

[0030] To illustrate by means of an example, suppose a form is going through a development process in different departments of a company. Each department may only be allowed to change particular elements of the form. For example, department A may be responsible for the form's corporate identity (fonts, layout, etc.), department B may be responsible for the form's interface to the application program, and department C may be responsible for developing the form content itself (texts, tables, etc.). Supposing the form represents an order, for example, there could be a form class (610) "Order" with three form types (620) "Corporate Identity", "Application Interface" and "Form Development". Form type "Corporate Identity" may contain special form rules (630) so form builder 510 only allows the change of the form layout and the styles for the texts. Form type "Application Interface" may contain special form rules so form builder 510 only allows changes to the form interface. Finally, form type "Form Development" may contain form rules so form builder 510 allows changes to everything except what the previous form rules are allowing. Using this embodiment, departments may hand over the form to the next department in the development process by changing the form type for the next department.

[0031] A system administrator may define access rights based on any distinction, such as user id, job title, department code, position in the corporate hierarchy, etc.

[0032] Several embodiments of the invention are specifically illustrated and/or described herein. However, it will be appreciated that modifications and variations of the invention are covered by the above teachings and within the purview of the appended claims without departing from the spirit and intended scope of the invention.